Progressive Cognitive Pipeline

An Ensemble Learning System for Conversational AI

By Jason Morrissette

J@4morr.com

10/19/2025

Ensemble Learning Overview

The Joshua Ecosystem Context
A Conversational AI Architecture

- The Joshua Ecosystem:
  - Compose of MADs (Multipurpose Agentic Duos)
  - MADs communicate through natural language conversations
  - Conversations stored permanently- communication and system memory
  - No APIs, no protocols—just dialogue
  - 16 patent-pending works, PCP among them
- Each MAD has:
  - Action Engine (Body): Executes domain tasks
  - Thought Engine (Brain): Cognitive processing with PCP

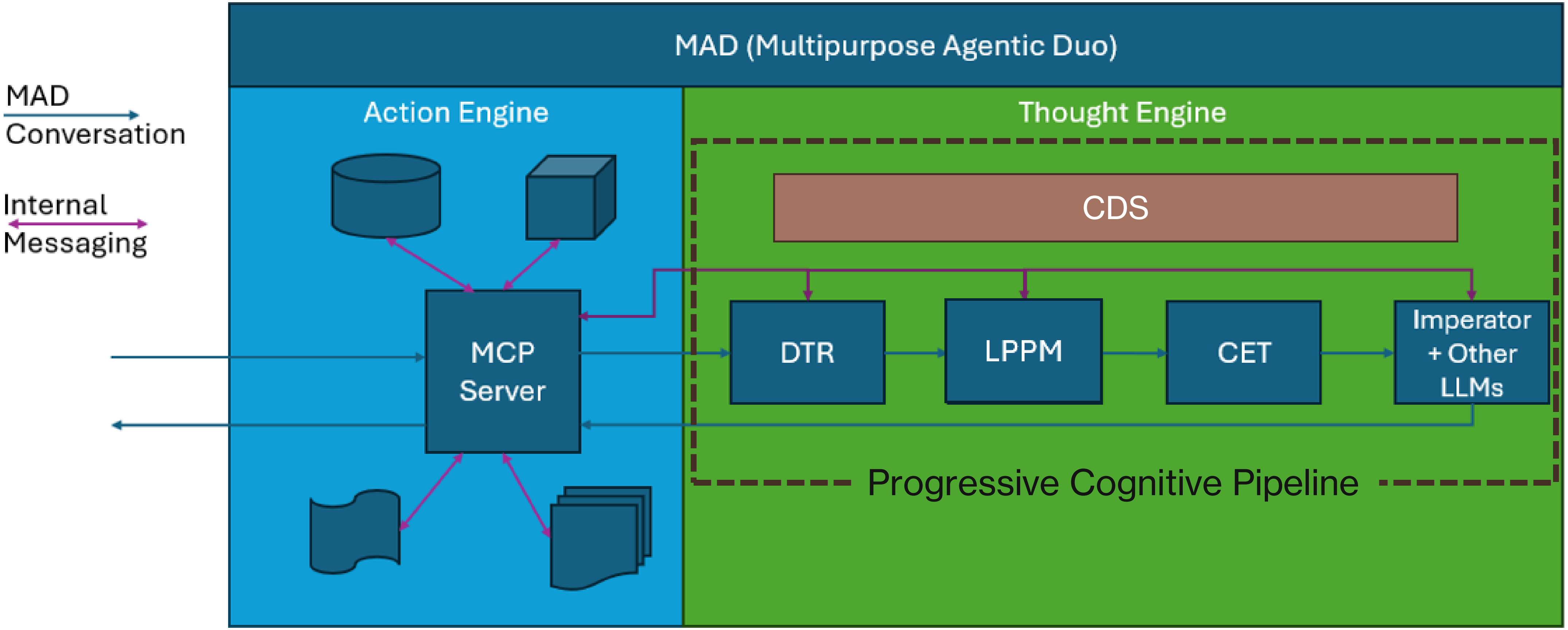
- Complete documentation: https://rmdevpro.github.io/rmdev-pro/

The Challenge With 100% Conversation

- MADs communicate exclusively through natural language conversations.
- No protocols are defined ahead of time
- Every interaction – whether a simple status check or complex collaborative reasoning – occurs through dialogue that is permanently stored as the system's memory.
- You can't use an LLM to process every system transaction.
    - Impractical
    - Slow
    - Expensive

Progressive Cognitive Pipeline

The Heart of the Thought Engine



Progressive Cognitive Pipeline Components

- Tier 1 - Decision Tree Router (DTR):
  - Machine learning classifier providing reflexive routing in microseconds. Handles deterministic commands, structured data, and learned routing patterns without semantic processing.
- Tier 2 - Learned Prose-to-Process Mapper (LPPM):
  - Neural network providing process orchestration in milliseconds and executes learned multi-step workflows that don't require creative reasoning. The LPPM compiles conversational workflows from observed patterns.
- Tier 3 - Context Engineering Transformer (CET):
  - Transformer network providing context optimization in hundreds of milliseconds. Assembles optimal context from multiple sources for LLM efficiency. The CET optimizes conversation history assembly for necessary reasoning.
- Tier 4 - Imperator (LLM):
  - Full semantic reasoning in seconds through API integration with an LLM. It handles novel situations, creative problem-solving, and genuine understanding requiring large language model capabilities.
- Tier 5 - Cognitive Recommendation System (CRS):
  - Metacognitive validation layer observing decisions across all tiers. Provides advisory recommendations questioning assumptions, suggesting alternatives, identifying capability gaps, and requesting consultation—without blocking execution.

PCP Sequential Cascade

Cognitive Stratification for Conversations

Tier	ML System	Speed	Handles
DTR	Machine learning classifier	Microseconds	"OK", "STATUS: COMPLETE" (60-80%)
LPPM	Neural network	Milliseconds	"Generate report" patterns (10-25%)
CET	Transformer network	100ms	Context optimization + parallelism
Imperator	LLM (API integration)	Seconds	Novel dialogues (<5% when trained)
CRS	Metacognitive validation	Continuous	Advisory oversight ("super ego")

Data And Learning

LLM Teaming And LLM Agile Case Study

LLMs Working Together for Extraordinary Results

The Agile LLM Approach:

- 5 LLMs working in parallel on 52 specifications
  - DeepSeek-R1, GPT-4, Claude, Gemini, Grok
- 7-model consensus review panel for quality
- Collaborative intelligence through multi-agent coordination

What Emerged from Teaming:

- Models collectively identified inefficiency patterns
- Democratic decision-making: 7-model unanimous agreement
- Strategic planning: Chose to regenerate for long-term consistency
- Delta format discovery: 76% token reduction
- Context parallelism: 19× speedup from batching

Results Through Collaboration:

- 3,467× speedup over human baseline
- 83% unanimous approval (quality through consensus)
- 18 minutes for 52 professional specifications
- Avoided drift and hallucination

Key Insight:

"Multiple diverse LLMs teaming together produce emergent intelligence

beyond any single model – strategic thinking, autonomous optimization, and

collaborative problem-solving“

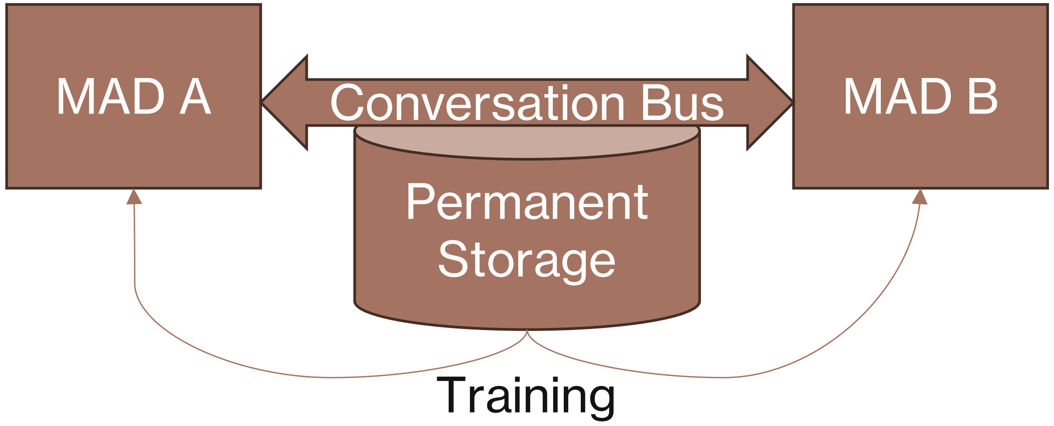
See appendix A of Joshua thesis paper: https://rmdevpro.github.io/rmdev-pro/projects/1_joshua/

LLM Teaming and Consultation

- LLM Teaming is use throughout the Joshua ecosystem, but is particularly useful in the PCP
- Cognitive Recommendation System is able to suggest that the Imperator seek consultation from out LLMs to
  - Improve data quality
  - Improve creativity
  - Reduce drift
  - Remove Hallucination

How MADs Learn from Conversation

The Conversation Bus as Memory



- **Key Innovation:**
  - Every conversation stored forever
  - MADs learn from their dialogue history
  - Cross-MAD learning through observed conversations
  - System improves through communication
  - **MADs teach themselves and each other through conversation**

Real Data Foundation

From Thousands of Actual Conversations

- Data Sources:
  - Developer-to-LLM dialogues during Joshua development
  - Inter-LLM communications (Agile LLM methodology)
  - Multiple formats: text files, databases, tool logs
- Processing Challenge:
  - Interleaved conversations (A→C→A→D→E pattern)
  - Missing timestamps requiring inference
  - Session IDs unreliable – discarded after analysis
- Hybrid Data Cleansing / Processing Approach:
  - Traditional Data Cleansing: Database deduplication, timestamp interpolation
  - Semantic ETL: LLMs parse interleaved dialogues
  - Result: Clean training corpus with conversation ID, text, timestamp, tags

LLM Based Bootstrap Training Data

- Along with real data synthetic data will be generated
- As conversation is the core of the systems, LLMs are perfectly placed to generate synthetic training data.
- LLMs will be used to create system requests and other forms of conversational traffic through the system

Implementation

Implementation Strategy
Versioned Deployment Within MADs

- Version 1 (Month 1): Conversational Baseline
  - 75% of the way through month 1
  - Deploy Imperator in all MAD Thought Engines
  - Build conversation history
  - Currently completing V1 deployment
  - PCP Design Documents Completed
- Version 2 (Month 2): Learn Patterns
  - LPPM learns from accumulated conversations
  - 5-10× improvement
- Version 3 (Month 3): Reflexive Routing
  - DTR classifies routine messages
  - 60-80% load reduction
- Version 4 (Month 4): Optimize Context
  - CET masters conversation history
  - Context parallelism optimization
- Version 5 (Month 5): Metacognitive Oversight
  - CRS "super ego" layer for quality
  - 50-100× total improvement